

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A method for accessing a data object stored in a first storage location in a computer, comprising:
determining whether another process is attempting to perform a transaction with the data object by determining whether an identifier (ID) of the data object is stored in a second lock object;
upon determining that the ID is not stored in the second lock object and that another process is not attempting to perform a transaction with the data object, storing an identifier (ID) the ID in [[a]] the second lock object, the ID being associated with a data object stored in a first storage location in a computer system;
determining whether another process is archiving the data object to a second storage location by determining whether the ID is contained stored in a first lock object and whether a link to a copy of the data object stored in the second storage location is assigned to the ID, wherein another process stores the ID in the first lock object and assigns the link to the ID while archiving the data object; and
upon determining that the ID is not stored in the first lock object, that the link is not assigned to the ID, and that another process is not archiving the data object,
performing at least one of a read operation and a write operation on the data object.

wherein the data object is locked if the ID is stored in at least one of the first lock object and the second lock object

~~if the ID is contained in the first lock object, determining whether a link to a second storage location having a copy of the data object is assigned to the ID in the first lock object, and~~

~~if the link is assigned to the ID, skipping performing a read and/or write access on the data object, and~~

~~if the link is not assigned to the ID, deleting the ID from the first lock object and performing the read and/or write access on the data object, and~~

~~if the ID is not contained in the first lock object, performing the read and/or write access on the data object.~~

2. (Previously Presented) The method of claim 1, wherein the first lock object is a file stored on a nonvolatile storage means.

3. (Previously Presented) The method of claim 1, wherein the first lock object comprises a table having a first column for the ID and a second column for the link of the ID to the second storage location.

4. (Currently Amended) The method of claim 1, wherein ~~[[each]]~~ the data object comprises one or more fields of one or more tables and wherein the ID comprises one or more key fields of the one or more tables.

5. (Previously Presented) The method of claim 4, wherein the link is a filename or a link to a file.

6. (Currently Amended) The method of claim 1, wherein the ~~first lock object~~ another process is created by a data moving process that stores the ID in the first lock object and assigns the link to the ID while archiving the data object.

7. (Previously Presented) The method of claim 1, wherein the second lock object is stored in a volatile storage means.

8. (Previously Presented) The method of claim 1, wherein the second lock object is a data array.

9. (Previously Presented) The method of claim 8, wherein the data array is one dimensional.

10. (Canceled).

11. (Currently Amended) A computer system for processing data, comprising:
memory means for storing program instructions;
input means for entering data;

storage means for storing data;

a processor responsive to the program instructions, wherein the program instructions comprise program code means for performing a method for accessing a data object ~~having an identifier (ID) and~~ stored in a first storage location, the method comprising:

determining whether another process is attempting to perform a transaction with the data object by determining whether an identifier (ID) of the data object is stored in a second lock object;

upon determining that the ID is not stored in the second lock object and that another process is not attempting to perform a transaction with the data object, storing the ID in [[a]] the second lock object;

determining whether another process is archiving the data object to a second storage location by determining whether the ID is contained stored in a first lock object and whether a link to a copy of the data object stored in the second storage location is assigned to the ID, wherein another process stores the ID in the first lock object and assigns the link to the ID while archiving the data object; and

upon determining that the ID is not stored in the first lock object, that the link is not assigned to the ID, and that another process is not archiving the data object, performing at least one of a read operation and a write operation on the data object,

wherein the data object is locked if the ID is stored in at least one of the first lock object and the second lock object

if the ID is contained in the first lock object, determining whether a link to a second storage location having a copy of the data object is assigned to the ID in the first lock object, and

if the link is assigned to the ID, skipping performing a read and/or write access on the data object, and

if the link is not assigned to the ID, deleting the ID from the first lock object and performing the read and/or write access on the data object, and

if the ID is not contained in the first lock object, performing the read and/or write access on the data object.

12. (Currently Amended) A computer readable storage medium comprising instructions executable by a processor for performing a method for accessing a data object stored in a first storage location, the method comprising:

determining whether another process is attempting to perform a transaction with the data object by determining whether an identifier (ID) of the data object is stored in a second lock object;

upon determining that the ID is not stored in the second lock object and that another process is not attempting to perform a transaction with the data object, storing an identifier (ID) the ID in [[a]] the second lock object, the ID being associated with a data object stored in a first storage location in a computer system;

determining whether another process is archiving the data object to a second storage location by determining whether the ID is contained stored in a first lock object

and whether a link to a copy of the data object stored in the second storage location is assigned to the ID, wherein another process stores the ID in the first lock object and assigns the link to the ID while archiving the data object; and

upon determining that the ID is not stored in the first lock object, that the link is not assigned to the ID, and that another process is not archiving the data object, performing at least one of a read operation and a write operation on the data object,

wherein the data object is locked if the ID is stored in at least one of the first lock object and the second lock object

~~if the ID is contained in the first lock object, determining whether a link to a second storage location having a copy of the data object is assigned to the ID in the first lock object, and~~

~~if the link is assigned to the ID, skipping performing a read and/or write access on the data object, and~~

~~if the link is not assigned to the ID, deleting the ID from the first lock object and performing the read and/or write access on the data object, and~~

~~if the ID is not contained in the first lock object, performing the read and/or write access on the data object.~~

13-14. (Canceled).

15. (Previously Presented) The computer readable storage medium of claim 12, wherein the first lock object is a file stored on a nonvolatile storage means.

16. (Previously Presented) The computer readable storage medium of claim 12, wherein the first lock object comprises a table having a first column for the ID and a second column for the link of the ID to the second storage location.

17. (Currently Amended) The computer readable storage medium of claim 12, wherein ~~[[each]]~~ the data object comprises one or more fields of one or more tables and wherein the ID comprises one or more key fields of the one or more tables.

18. (Previously Presented) The computer readable storage medium of claim 12, wherein the link is a filename or a link to a file.

19. (Currently Amended) The computer readable storage medium of claim 12, wherein the ~~first lock object~~ another process is ~~created by~~ a data moving process that stores the ID in the first lock object and assigns the link to the ID while archiving the data object.

20. (Previously Presented) The computer readable storage medium of claim 12, wherein the second lock object is stored in a volatile storage means.

21. (Previously Presented) The computer readable storage medium of claim 12, wherein the second lock object is a data array.

22. (Previously Presented) The computer readable storage medium of claim 21, wherein the data array is one dimensional.

23. (Currently Amended) A computer system for accessing a data object stored in a first storage location, comprising:

a processor responsive to program instructions;

means for determining whether another process is attempting to perform a transaction with the data object by determining whether an identifier (ID) of the data object is stored in a second lock object;

means for storing an identifier (ID) the ID in [[a]] the second lock object, the ID being associated with a data object stored in a first storage location upon determining that the ID is not stored in the second lock object and that another process is not attempting to perform a transaction with the data object;

means for determining whether another process is archiving the data object to a second storage location by determining whether the ID is contained stored in a first lock object and whether a link to a copy of the data object stored in the second storage location is assigned to the ID, wherein another process stores the ID in the first lock object and assigns the link to the ID while archiving the data object; and

means for performing at least one of a read operation and a write operation on the data object upon determining that the ID is not stored in the first lock object, that the link is not assigned to the ID, and that another process is not archiving the data object.

wherein the data object is locked if the ID is stored in at least one of the first lock object and the second lock object

~~if the ID is contained in the first lock object, determining whether a link to a second storage location having a copy of the data object is assigned to the ID in the first lock object, and~~

~~if the link is assigned to the ID, skipping performing a read and/or write access on the data object, and~~

~~if the link is not assigned to the ID, deleting the ID from the first lock object and performing the read and/or write access on the data object, and~~

~~if the ID is not contained in the first lock object, performing the read and/or write access on the data object.~~

24. (Previously Presented) The computer system of claim 23, wherein first lock object is a file stored on a nonvolatile storage means.

25. (Previously Presented) The computer system of claim 23, wherein the first lock object comprises a table having a first column for the ID and a second column for the link of the ID to the second storage location.

26. (Currently Amended) The computer system of claim 23, wherein [[each]] the data object comprises one or more fields of one or more tables and wherein the ID comprises one or more key fields of the one or more tables.

27. (Previously Presented) The computer system of claim 23, wherein the link is a filename or a link to a file.

28. (Currently Amended) The computer system of claim 23, wherein the first lock object another process is created by a data moving process that stores the ID in the first lock object and assigns the link to the ID while archiving the data object.

29. (Previously Presented) The computer system of claim 23, wherein the second lock object is stored in a volatile storage means.

30. (Previously Presented) The computer system of claim 23, wherein the second lock object is a data array.

31. (Previously Presented) The computer system of claim 30, wherein the data array is one dimensional.